基于任意分割的串行进位链规则获取的计算流程

程玉胜[1], 张佑生[2,3], 胡学钢[3], 章晓良[4]
(1. 安庆师范学院计算机与信息学院, 安徽安庆 246011; 2. 安徽三联学院计算机科学与技术系, 安徽合肥 230601;
3. 合肥工业大学计算机与信息学院, 安徽合肥 230009; 4. 合肥工业大学机械与汽车工程学院, 安徽合肥 230009)

摘 要: 分析了等价矩阵和联合决策矩阵规则提取算法对于大数据集低效性的根源. 提出了基于任意分割的规则获取方法和相应的串行进位链计算流程. 这种计算流程将大数据集上的规则获取, 转化为通过分割后多个智能体(子系统)及其智能体间数据共享的"并行+串行"的规则提取计算过程, 有效的解决了大数据集上规则获取问题. 复杂度分析表明该算法在效率上较现有的算法有显著的提高; 实例分析验证了该方法的可行性; 相应的对比实验表明这种计算流程对大数据集上的规则获取的实用性和高效性.

关键词: 粗糙集理论; 串行进位链; 智能体; 矩阵分块; 联合决策矩阵
中图分类号: TP18 文献标识码: A 文章编号: 0372-2112 (2009) 12-2797-06

Calculating Process with Serial Carry Chain for Rules Extraction Based on the Arbitrary Division

CHENG Yu-sheng[1], ZHANG You-sheng[2,3], HU Xue-gang[3], ZHANG Xiao-liang[4]
(1. *School of Computer and Information, Anqing Teachers College, Anqing, Anhui 246011, China;*
2. *Department of Computer science &Technology, Anhui Sanlian University, Hefei, Anhui 230601, China;*
3. *School of Computer Science, Hefei University of Technology, Hefei, Anhui 230009, China;*
4. *School of Mechanical and Automotive Engineering, Hefei University of Technology, Hefei, Anhui 230009, China*)

Abstract: Based on equivalence matrix and joint decision matrix, the reason of the existing algorithms inefficiency for rules extraction in massive data set is analyzed. The method of rules extraction and calculating process with serial carry chain based on the arbitrary division are presented. This process about the rules extraction will be changed into many agent (sub-systems) and inter-agent to share data by the "Parallel plus Serial" rule calculation, which can effectively improve the algorithm on the massive data set. Complexity analysis shows that the algorithm is more efficient than those existing algorithms. An example is used to illustrate the efficiency of the new algorithm. At last, experimental result shows that the calculating process with serial carry chain for rules extraction is not only efficient but also scalable.

Key words: rough sets theory; serial carry chain; agent; matrix block; joint decision matrix

1 引言

粗糙集理论作为一种新的处理不确定性的数学工具, 已经成功的应用于模式识别、数据挖掘等相关领域. 同其它数学工具相比, 该理论在不需要先验知识的情况下可以对信息系统中的冗余属性进行约简, 其中, 分辨矩阵约简方法最为常用[1,2]. 但是叶东毅指出该矩阵存在构造上的缺陷, 并改进了分辨矩阵定义及其相应的核属性求解方法[3].

然而, 当所研究的决策信息系统数据集相当大甚至是海量时, 文献[1~3]中的分辨矩阵算法就很难从这种大规模数据集上获取知识. 为了解决此问题, 常见的是在原有数据集上研究高效的属性约简算法, 如刘少辉[4]和徐章艳[5]分别从降低时间复杂度的角度研究了大数据集上的约简和规则求解方法; 或者是在子系统约简的基础上研究大系统的约简问题, 如 Bazan[6] 通过对大型决策信息系统进行多次抽样, 把复杂决策信息系统的约简问题转化为若干子决策信息系统的约简交集, 提出动态约简概念和广义动态约简思想体系; 由此王加阳[7]把约简精度系数引入到抽样估计中, 提出抽样计算的新方法; M. Kryszkiewicz[8] 研究了怎样利用现有子系统的约简求复合系统的约简问题以及邓大勇[9]提出了连接两个信息系统的分辨矩阵定义和核属性求解方法等. 这些研究在一定程度上满足了从小样本到大数集上进行知识

收稿日期: 2008-10-07; 修回日期: 2009-06-08
基金项目: 安徽省自然科学基金(No. 070412061)

获取的需求.

但是这些方法都是基于分辨矩阵的, 容易造成规则损失, 由此Guan[10]提出了等价矩阵定义, 该定义将条件属性集和决策属性所包含的等价关系浓缩到等价矩阵中, 为约简和规则获取提供了一个新思路. 文香军[11]以及谭天乐[12]在矩阵运算的基础上, 提出等价矩阵的规则提取算法; 黄兵[13]将等价矩阵思想拓展到不完备信息系统中, 给出了基于相容矩阵和决策分配矩阵的规则提取方法. 这些矩阵算法的优点在于将属性约简和属性值约简集成在一起, 提高了算法的效率, 同时也避免了约简属性集外规则的损失. 但是, 当样本数很大时, 由于等价矩阵算法对样本数非常敏感, 因此这些算法应用受到一定的限制.

在联合决策矩阵的基础上, 文献[14]提出了基于决策类分割的串行进位链规则获取的矩阵分块算法(FMRE)[14], 但是决策类数目对这种算法性能影响较大, 因此FMRE算法不适合于从更大规模的数据集提取规则. 基于此, 本文提出了基于任意分割的串行进位链规则提取计算流程, 这种计算流程将大数据集上的规则获取, 转化为通过分割后多个智能体(子系统)及其智能体间数据共享的"并行+串行"的规则提取计算过程, 有效的解决了大数据集上规则获取问题.

2 联合决策矩阵及其FMRE算法局限性[14]

基于等价矩阵的规则获取算法不仅需要分别生成条件属性和决策属性的等价矩阵[10~13], 而且矩阵的规模也限制了其在大数据集上的有效应用. 但是, 基于联合决策矩阵的概念和相应的矩阵分块算法[14], 在一定程度上为大数据集上的知识获取提供了可能.

定义1 $S=(U,A,V,f)$是一个决策系统, $F\subseteq A$, $U=\{x_1,x_2,\cdots,x_n\}$, 则等价矩阵 $\boldsymbol{M}_F(\boldsymbol{S})$ 定义为:

$$\boldsymbol{M}_F(\boldsymbol{S})=[r_{i,j}]_{n\times n}$$

其中$r_{i,j}$表示等价矩阵中第i行第j列的元素, 定义为:

$$r_{i,j}=\begin{cases}1 & x_iE_Fx_j\\ 0 & \text{otherwise}\end{cases},\ i,j=1,2,\cdots,n$$

其中E_F是等价关系, 定义为:

$$xE_Fy\Leftrightarrow\forall b\in F,x,y\in U,f(x,b)=f(y,b)$$

定义2 对于决策系统 $S_1=(U_1,C\cup\{d\},V)$和$S_2=(U_2,C\cup\{d\},V)$, 其中, $U_1=\{x_1,x_2,\cdots,x_\gamma\}$, $U_2=\{y_1,y_2,\cdots,y_\beta\}$, $C=\{c_1,c_2,\cdots,c_m\}$, $B\subseteq C\cup\{d\}$, 则S_1, S_2间的等价矩阵定义为:

$$\boldsymbol{M}_B(\boldsymbol{S}_1,\boldsymbol{S}_2)=[m_{i,j}]_{\gamma\times\beta}$$

其中, $m_{i,j}$表示等价矩阵中第i行第j列的元素, 定义为:

$$m_{i,j}=\begin{cases}1, & x_iE_By_j\wedge x_i\in U_1\wedge y_j\in U_2\\ 0, & \text{otherwise}\end{cases}$$

其中: $i=1,2,\cdots,\gamma$; $j=1,2,\cdots,\beta$.

根据决策属性求出所有决策类, 记为 U/E_D: $U/E_D=\{D_1,D_2,\cdots,D_p\}$; 然后根据决策类的个数$p$将原决策系统分割为$p$个子系统$S_i=(D_i,A,V,f)$, 其中$D_i\in U/E_D$, $i=1,2,\cdots,p$. 按照$U'=D_1\cup D_2\cup\cdots\cup D_p$重新排列论域$U$得$S'=(U',A,V,f)$. 则联合决策矩阵定义为:

定义3 决策系统$S=(U,A,V,f)$被分割为p个子系统$S_i=(D_i,A,V,f)$, $B\subseteq C$; 原决策系统S转化为$S'=(U',A,V,f)$. 则决策系统S'的联合决策矩阵$\boldsymbol{M}_B(\boldsymbol{S}')$可以用其$p$个子系统$S_i$的等价矩阵和$S_i$、$S_j$间的等价矩阵来表达, 即:

$$\boldsymbol{M}_B(\boldsymbol{S}')=\begin{bmatrix}\boldsymbol{M}_B(\boldsymbol{S}_1,\boldsymbol{S}_2) & \boldsymbol{M}_B(\boldsymbol{S}_1,\boldsymbol{S}_3) & \cdots & \boldsymbol{M}_B(\boldsymbol{S}_1,\boldsymbol{S}_{p-1}) & \boldsymbol{M}_B(\boldsymbol{S}_1,\boldsymbol{S}_p)\\ & \boldsymbol{M}_B(\boldsymbol{S}_2,\boldsymbol{S}_3) & \cdots & \boldsymbol{M}_B(\boldsymbol{S}_2,\boldsymbol{S}_{p-1}) & \boldsymbol{M}_B(\boldsymbol{S}_2,\boldsymbol{S}_p)\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ & & \cdots & & \boldsymbol{M}_B(\boldsymbol{S}_{p-1},\boldsymbol{S}_p)\end{bmatrix}$$

联合决策矩阵优于等价矩阵, 这是因为:

(1)联合决策矩阵中子系统S_i的等价矩阵$\boldsymbol{M}_B(\boldsymbol{S}_i)$不必生成;

(2)联合决策矩阵可以采取串行进位链计算流程来提取每个子系统中的规则.

不足之处是两类决策系统是最常见的[15], 因此FMRE算法受到决策类数目的限制, 所以该算法不适应更大规模的数据集. 为此, 下面给出基于任意分割的串行进位链规则获取矩阵分块方法.

3 PSRE获取方法及其串行进位链计算流程

3.1 PSRE获取方法

为了描述串行进位链规则提取算法, 首先介绍向量"小于等于"运算.

定义4 设$\boldsymbol{V}_1=\{v_{11},v_{12},\cdots,v_{1n}\}$和$\boldsymbol{V}_2=\{v_{21},v_{22},\cdots,v_{2n}\}$是两个$1\times n$的向量, 定义[13]:

$$\boldsymbol{V}_1\leqslant\boldsymbol{V}_2\Leftrightarrow v_{1i}\leqslant v_{2i},\ i=1,2,\cdots,n$$

定理1 假设大数据集$S=(U,C\cup\{d\},V)$被分割成2个子系统, 记为$S_i=(U_i,C\cup\{d\},V)$ $(i=1,2)$, 其中$\bigcup_{i=1}^{2}U_i=U$, $U_1\cap U_2=\Phi$, $B\subseteq C$. $\boldsymbol{M}_B(\boldsymbol{S}_i)=(\vec{\boldsymbol{V}}_{B1},\vec{\boldsymbol{V}}_{B2},\cdots,\vec{\boldsymbol{V}}_{B|U_i|})^{\mathrm{T}}$, $\boldsymbol{M}_{\{d\}}(\boldsymbol{S}_i)=(\vec{\boldsymbol{V}}_{D1},\vec{\boldsymbol{V}}_{D2},\cdots,\vec{\boldsymbol{V}}_{D|U_i|})^{\mathrm{T}}$, $(i=1,2)$, 分别表示子系统关于B, $\{d\}$的等价矩阵. $\boldsymbol{M}_B(\boldsymbol{S}_i,\boldsymbol{S}_j)=(\vec{\boldsymbol{V}}_{B1},\vec{\boldsymbol{V}}_{B2},\cdots,\vec{\boldsymbol{V}}_{B|U_i|})^{\mathrm{T}}$, $\boldsymbol{M}_{\{d\}}(\boldsymbol{S}_i,\boldsymbol{S}_j)=(\vec{\boldsymbol{V}}_{D1},\vec{\boldsymbol{V}}_{D2},\cdots,\vec{\boldsymbol{V}}_{D|U_i|})^{\mathrm{T}}$ $(j\neq i;\ i,j=1,2)$分别表示两个子信息系统之间关于B, $\{d\}$的等价矩阵. 对于$k\in[1,|U_i|]$, 如果$\exists(\vec{\boldsymbol{V}}_{Bk}\leqslant\vec{\boldsymbol{V}}_{Dk})\wedge(\vec{\boldsymbol{V}}_{Bk}\leqslant\vec{\boldsymbol{V}}_{Dk})$ $(j\neq i;\ j=1,2)$, 则子系统S_i中有一条决策规则$\bigwedge_{b\in B,x_k\in U_i}(b=b(x_k))\Rightarrow d=d(x_k)$.

证明　假设向量 $\boldsymbol{V}^i_{Bk}=(v^i_{Bk_1}, v^i_{Bk_2}, \cdots, v^i_{Bk_{|U_i|}})$, $\boldsymbol{V}^i_{Dk}=(v^i_{Dk_1}, v^i_{Dk_2}, \cdots, v^i_{Dk_{|U_i|}})$.

首先证明在子系统 S_i 中, 如果 $\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk}$, 则 $R: \bigwedge_{b\in B, x_k\in U_i}(b=b(x_k))\Rightarrow d=d(x_k)$ 是确定性规则.

反证法　如果 $\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk}$, R 在子系统 S_i 中是矛盾规则, 即表明 R 规则的前件相同, 后件不同; 因此根据等价矩阵的定义, 在向量 $\boldsymbol{V}^i_{Bk}$ 中必 $\exists v^i_{Bk_f}=1$, 在向量 $\boldsymbol{V}^i_{Dk}$ 中有 $v^i_{Dk_f}=0$, 即 $v^i_{Bk_f}>v^i_{Dk_f}$, 所以 $\boldsymbol{V}^i_{Bk}>\boldsymbol{V}^i_{Dk}$, 这与 $\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk}$ 是矛盾的, 故当 $\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk}$ 时, 在子系统 S_i 中 R 是确定性规则;

其次证明当 $\boldsymbol{V}^j_{Bk}\leqslant\boldsymbol{V}^j_{Dk}$ 成立时, 从子系统 S_i 中得出的规则 R 如果与子系统 $S_j(j\neq i;\ j=1, 2)$ 并不矛盾的话, 即表明 R 是大数据集上的一条确定性规则.

任意取一个子系统 S_j, 假设 R 与 S_j 中一个样本 $x_t\in U_j$ 描述的规则矛盾, 则必有 $b(x_k)=b(x_t)$, $\forall b\in B$, 而 $d(x_k)=d(x_t)$, 这样在向量 $\boldsymbol{V}^j_{Bk}$ 中 $\exists v^i_{Bk_t}=1$, 在向量 $\boldsymbol{V}^j_{Dk}$ 中有 $v^i_{Dk_t}=0$, 即 $v^i_{Bk_t}>v^i_{Dk_t}$, 所以 $\boldsymbol{V}^i_{Bk}>\boldsymbol{V}^i_{Dk}$ 与题设 $\boldsymbol{V}^j_{Bk}\leqslant\boldsymbol{V}^j_{Dk}$ 矛盾, 故 R 是大数据集上的一条确定性规则.

证毕

定理 1 给出了数据集分割成 2 个子系统后的规则提取方法. 如果数据集分割成两个子系统后, 其子系统的样本数仍然很多, 则需要进一步分割. 下面的推论给出了大数据集分割成 p 个子系统的规则获取方法.

推论 **1**　假设大数据集 $S=(U, C\cup\{d\}, V)$ 被分割成 p 个子系统, 记为 $S_i=(U_i, C\cup\{d\}, V)$, $i=1, 2, \cdots, p$, 其中 $\bigcup_{i=1}^{p}U_i=U$, $U_i\cap U_j=\Phi(i\neq j;\ i, j=1, 2, \cdots, p)$, $B\subseteq C$; $\boldsymbol{M}_B(\boldsymbol{S}_i)=(\boldsymbol{V}^i_{B1}, \boldsymbol{V}^i_{B2}, \cdots, \boldsymbol{V}^i_{B|U_i|})^{\mathrm{T}}$, $\boldsymbol{M}_{\{d\}}(\boldsymbol{S}_i)=(\boldsymbol{V}^i_{D1}, \boldsymbol{V}^i_{D2}, \cdots, \boldsymbol{V}^i_{D|U_i|})^{\mathrm{T}}$, $\boldsymbol{M}_B(\boldsymbol{S}_i, \boldsymbol{S}_j)=(\boldsymbol{V}^j_{B1}, \boldsymbol{V}^j_{B2}, \cdots, \boldsymbol{V}^j_{B|U_i|})^{\mathrm{T}}$, $\boldsymbol{M}_{\{d\}}(\boldsymbol{S}_i, \boldsymbol{S}_j)=(\boldsymbol{V}^j_{D1}, \boldsymbol{V}^j_{D2}, \cdots, \boldsymbol{V}^j_{D|U_i|})^{\mathrm{T}}(j\neq i;\ i, j=1, 2, \cdots, p)$, 分别表示子系统 S_i 以及 S_i, S_j 间条件属性集 B, 决策属性 $\{d\}$ 的等价矩阵. 对于 $k\in[1, |U_i|]$, 如果 $\exists(\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk})\wedge(\boldsymbol{V}^j_{Bk}\leqslant\boldsymbol{V}^j_{Dk})$, 则子系统 S_i 中有一条决策规则 $\bigwedge_{b\in B, x_k\in U_i}(b=b(x_k))\Rightarrow d=d(x_k)$.

推论 **2**　当大数据集 $S=(U, C\cup\{d\}, V)$ 未被分割, 即 $p=1$ 时, 对于 $k\in[1, |U|]$, 如果 $\exists(\boldsymbol{V}^i_{Bk}\leqslant\boldsymbol{V}^i_{Dk})$, 则系统 S 中有一条决策规则 $\bigwedge_{b\in B, x_k\in U}(b=b(x_k))\Rightarrow d=d(x_k)$.

推论 2 实际上是两类决策系统规则获取方法特例, 其相关结论已经应用到完备信息系统[15]和不完备信息系统中的规则提取上[13, 16].

3.2　大数据集任意分割原则

假设大数据集为 S, 样本数为 $n=|U|$; 分割后的两个子系统分别为 S_1、S_2, 其样本数分别为 $\gamma=|U_1|$, $\beta=|U_2|$; 显然 $\gamma+\beta=n$. 在等价矩阵的规则提取算法中, 1 矩阵是整个算法的基础. 表 5 中的实验数据已经表明如果 1 矩阵的空间过大, 可能导致矩阵算法终止; 因此降低 1 矩阵规模是必要的, 对于大数据集更需要这样. 两个信息系统间的等价矩阵定义实际上提供了通过分割大数据集来降低 1 矩阵规模的方法. 下面讨论如何选择分割点使得 1 矩阵规模最小.

因为分割后 $\boldsymbol{M}_B(\boldsymbol{S}_1)$ 占用的空间为 γ^2, $\boldsymbol{M}_B(\boldsymbol{S}_2)$ 占用的空间为 β^2, 系统间的矩阵 $\boldsymbol{M}_B(\boldsymbol{S}_1, \boldsymbol{S}_2)$ 占用的空间为 $\gamma\times\beta$, 所以 1 矩阵占用的空间为 $(\gamma^2+\beta^2+\gamma\times\beta)\times(|C|+1)$; 而 $\gamma+\beta=n$, 通过极值法, 当 $\gamma=\beta=n/2$ 时, $\gamma^2+\beta^2+\gamma\times\beta$ 最小值为 $3n^2/4$, 所以通过均等分割大数据集, 整个 1 矩阵占用的空间达到最小, 为 $\frac{3n^2}{4}(|C|+1)$.

可见, 将大数据集分割成 2 个子系统时, 均等分割是最好的策略. 同样, 均等分割对于分割为 p 个子系统也是最好的方法.

3.3　串行进位链计算流程

以分割为 3 个子系统为例, "串行进位链" 规则获取计算流程, 如图 1 所示. 其中, $\boldsymbol{M}(\boldsymbol{S}_i, \boldsymbol{S}_j)$ 表示两个子系统间的分辨矩阵或等价矩阵, 记为 $\boldsymbol{M}_{ij}$, 红色线表示"进位链", T 表示矩阵转置.

图 1 表明, 在子系统中的规则获取, 可以同时("并行")计算图中的 $\boldsymbol{M}_{11}$, $\boldsymbol{M}_{12}$, $\boldsymbol{M}_{13}$, $\boldsymbol{M}_{22}$, $\boldsymbol{M}_{23}$, $\boldsymbol{M}_{33}$; 而对于子系统 S_2, 可以通过"进位链"结构"串行"获取 $\boldsymbol{M}_{21}$, 对于子系统 S_3, 可以通过"进位链"结构"串行"获取 $\boldsymbol{M}_{31}$, $\boldsymbol{M}_{32}$. 可以预见, 这种"并行+串行"方法实现了部分矩阵的共享, 减少了重复劳动, 将能有效的解决更大规模数据集上的规则获取问题.

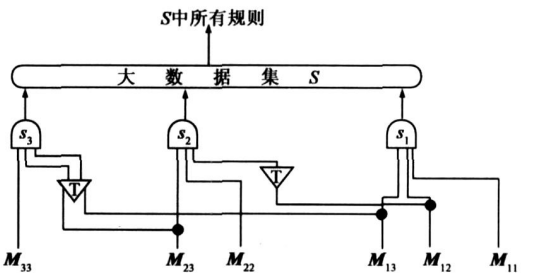

图1　大数据集上"串行进位链"计算流程(p=3)

4　PSRE 算法

4.1　算法描述

假设大数据集 S 被分割成 p 个子系统 $S_i=(U_i, C\cup\{d\}, V)$, $i=1, 2, \cdots, p$, $\gamma_i=|U_i|$, $i=1, 2, \cdots, p$. 为了避免在生成一条规则后, 在提取涉及更多属性规则时,

产生与之存在包含关系的冗余规则, 需要设置相应属性规则生成标志, 下面算法中, "*"表示相应规则已经生成. 基于任意分割的"串行+ 并行"规则提取矩阵算法(Parallel plus Serial for Rule Extraction, PSRE) 描述如下:

算法 PSRE:

输入: $S=(U, C\cup\{d\}, V)$ 以及分割数 p

输出: $S=(U, C\cup\{d\}, V)$ 的所有决策规则

Step1 将 S 分割为 p 个子系统 $(S_1, S_2, \cdots, S_p)$

Step2 提取子系统 S_i 中的规则:

$Rule(S_i)=agent_i(S_i, S_{i+1}, \cdots, S_p)$, $i=1, 2, \cdots, p$

分割后的每个子系统规则获取可以看成是一个智能体 *agent*, 因此获取 S_i 系统中的规则可以看成 $agent_i$ "并行" 计算 $\boldsymbol{M}_{ii}$ 到 $\boldsymbol{M}_{ip}$, 通过"串行" 进位链接受 $agent_1$ 到 $agent_{i-1}$ 获取的矩阵 $\boldsymbol{M}_{1i}$ 到 $\boldsymbol{M}_{i-1, i}$.

算法描述如下:

算法 $agent_i$:

输入参数: $S_i, S_{i+1}, \cdots, S_p$

返回参数: 子系统 S_i 中的规则

Step1 计算 $\boldsymbol{M}_D(\boldsymbol{S}_i)$, $\boldsymbol{M}_D(\boldsymbol{S}_i, \boldsymbol{S}_i)$, $j\neq i, j=1, 2, \cdots, p$

Step2 当 $k\leqslant m=|C|$ 时, 做:

Step3 计算等价矩阵 $\boldsymbol{M}_B^k(\boldsymbol{S}_i)$ 和系统间等价矩阵 $\boldsymbol{M}_B^k(\boldsymbol{S}_i, \boldsymbol{S}_j)$, $j\neq i, j=1, 2, \cdots, p$.

Step4 根据定理 1 或其推论 1 生成子系统 S_i 中的规则, 并在等价矩阵最后一列相应的行设置规则生成标志位'*'

Step5 如果 $\boldsymbol{M}_B^k(\boldsymbol{S}_i)$ 最后一列全为'*', 结束; 否则 $k=k+1$, 转 Step2.

在子系统 S_i 规则获取的流程中, 对某个 k 值, 需要计算 $2\times p$ 个等价矩阵. 进一步分析不难发现, 当 $j<i$ 时, 根据 $\boldsymbol{M}_B(\boldsymbol{S}_2, \boldsymbol{S}_1)=\boldsymbol{M}_B(\boldsymbol{S}_1, \boldsymbol{S}_2)^{\mathrm{T}}$ 对称性可以获得相应的等价矩阵, 因此在算法 $agent_i$ 中实际上只需要计算 $2\times(p-i+1)$ 个等价矩阵.

4.2 复杂度分析

设将大数据集 S(样本数为 $n=|U|$) 分割为 p 个子信息系统(其样本数为 $\gamma_i=|U_i|$), $i=1, 2, \cdots, p$.

这样, 可将大数据集上的规则提取转化为 p 个子信息系统规则提取的并行计算, 因此 PSRE 算法的时间复杂度为各个子信息系统计算中复杂度最大值. 如果采用均等分割原则, 则为 $O(2^{|C|}n^2/p^2)$. 特别地: 当 $p=4$ 时, PSRE 算法的时间复杂度是 $O(2^{|C|-4}n^2)$, 与文献[11]基于 RAVT 的矩阵算法 RMC 时间复杂度 $O(2^{|C|-1}n^2)$ 相当.

对于空间复杂度, 只需要计算 $agent_i$ 中 $\boldsymbol{M}_D(\boldsymbol{S}_i)$, $\boldsymbol{M}_D(\boldsymbol{S}_i, \boldsymbol{S}_i)$, $\boldsymbol{M}_B^k(\boldsymbol{S}_i)$ 和 $\boldsymbol{M}_B^k(\boldsymbol{S}_i, \boldsymbol{S}_j)$ 的 1 矩阵占用空间, 这些矩阵所占空间分别为 γ_i^2、$\gamma_i\sum_{j=1, j\neq i}^{p}\gamma_j=\gamma_i(n-\gamma_i)$、$\gamma_i^2$ 和 $\gamma_i(n-\gamma_i)$, 共为 $n\gamma_i+|C|n\gamma_i$. 如果是均等分割, 则为 $(|C|+1)n^2/p$, 约为 RMC 矩阵算法所占空间 $(|C|+1)n^2$ 的 $1/p$. "out of memory" 问题得到缓解.

4.3 实例分析

根据均等分割原则, 将表 1 决策系统分割成两个子系统, 如表 2、表 3 所示.

表 1 决策信息系统 S

U	a	b	c	D
$x1$	2	2	2	2
$x2$	2	2	3	3
$x3$	2	3	2	4
$y1$	2	3	3	3
$y2$	3	2	2	4
$y3$	3	2	3	2

表 2 决策子系统 S_1

U	a	b	c	D
$x1$	2	2	2	2
$x2$	2	2	3	3
$x3$	2	3	2	4

表 3 决策子系统 S_2

U	a	b	c	D
$y1$	2	3	3	3
$y2$	3	2	2	4
$y3$	3	2	3	2

计算 $\boldsymbol{M}_D(\boldsymbol{S}_1)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$, $\boldsymbol{M}_D(\boldsymbol{S}_1, \boldsymbol{S}_2)=\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}$.

下面给出提取子系统 S_1 规则过程:

当 $k=1$ 时, 分别计算 $\boldsymbol{M}_B^1(\boldsymbol{S}_1)$, $\boldsymbol{M}_B^1(\boldsymbol{S}_1, \boldsymbol{S}_2)$:

$$\boldsymbol{M}_a^1(\boldsymbol{S}_1)=\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}, \boldsymbol{M}_b^1(\boldsymbol{S}_1)=\begin{bmatrix}1&1&0\\1&1&0\\0&0&1\end{bmatrix},$$

$$\boldsymbol{M}_c^1(\boldsymbol{S}_1)=\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix}, \boldsymbol{M}_a^1(\boldsymbol{S}_1, \boldsymbol{S}_2)=\begin{bmatrix}1&0&0\\1&0&0\\1&0&0\end{bmatrix},$$

$$\boldsymbol{M}_b^1(\boldsymbol{S}_1, \boldsymbol{S}_2)=\begin{bmatrix}0&1&1\\0&1&1\\1&0&0\end{bmatrix}, \boldsymbol{M}_c^1(\boldsymbol{S}_1, \boldsymbol{S}_2)=\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix},$$

第一层没有规则生成. 下面计算 $k=2$ 时对应矩阵:

$$\boldsymbol{M}_{ab}^2(\boldsymbol{S}_1)=\begin{bmatrix}1&1&0\\1&1&0\\0&0&1\end{bmatrix}, \boldsymbol{M}_{ac}^2(\boldsymbol{S}_1)=\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix},$$

$$\boldsymbol{M}_{bc}^2(\boldsymbol{S}_1)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}, \boldsymbol{M}_{ab}^2(\boldsymbol{S}_1, \boldsymbol{S}_2)=\begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix},$$

$M^2_{ac}(S_1, S_2)=\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix}$, $M^2_{bc}(S_1, S_2)=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}$,

因为 $M^2_{ac}(S_1)$ 的第 2 行等于 $M_D(S_1)$ 的第 2 行且 $M^2_{ac}(S_1, S_2)$ 的第 2 行等于 $M_D(S_1, S_2)$ 的第 2 行, 所以得到一条规则: $a=2 \wedge c=3 \Rightarrow D=3$.

$M^2_{bc}(S_1)$ 的第 3 行等于 $M_D(S_1)$ 的第 3 行且 $M^2_{bc}(S_1, S_2)$ 的第 3 行小于 $M_D(S_1, S_2)$ 的第 3 行, 得到一条规则: $b=3 \wedge c=2 \Rightarrow D=4$.

同时修改相应矩阵:

$M^2_{ac}(S_1)=\begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix}*$, $M^2_{bc}(S_1)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}*$,

$k=3$ 时对应矩阵: $M^3_{abc}(S_1)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}*$, 因为 $M^3_{abc}(S_1)$ 的第 1 行等于 $M_D(S_1)$ 的第 1 行且 $M^3_{abc}(S_1, S_2)$ 的第 1 行等于 $M_D(S_1, S_2)$ 的第 1 行, 所以得到一条规则: $a=2 \wedge b=2 \wedge c=2 \Rightarrow D=2$; 修改相应矩阵:

$M^3_{abc}(S_1)=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}*$, 子系统 1 算法结束;

同样, 通过第 2 个子系统计算, 提取的规则为: $a=3 \wedge c=2 \Rightarrow D=4$, $a=3 \wedge c=3 \Rightarrow D=2$, $b=3 \wedge c=3 \Rightarrow D=3$. 故 PSRE 算法从表 1 中提取 6 条规则, 如表 4 所示, 与文献[11]的计算结果一致但矩阵规模要小.

表 4　规则表

	规则
$R1$	$a=2 \wedge b=2 \wedge c=2 \Rightarrow D=2$
$R2$	$a=2 \wedge c=3 \Rightarrow D=3$
$R3$	$b=3 \wedge c=2 \Rightarrow D=4$
$R4$	$b=3 \wedge c=3 \Rightarrow D=3$
$R5$	$a=3 \wedge c=2 \Rightarrow D=4$
$R6$	$a=3 \wedge c=3 \Rightarrow D=2$

表 5　分割数 p 对时空复杂度影响对比

样本	条件数	样本数	1 矩阵空间/bytes			1 矩阵生成时间/s		
			$p=1$	$p=2$	$p=4$	$p=1$	$p=2$	$p=4$
Heart-statlog	13	270	8.165M	4.08M	2.07M	3.609	1.641	0.875
Credit-g	20	1000	168M	84M	42M	50.719	33	16.484
KR-vs-KP	20	3196	Out of memory	1716.02M	429.01M	×	444.828	159.984

注: 实验环境: Windows XP, MATLAB 6.5, 1.6GHz

5　实验结果

在 $p=1, 2, 4$ 三种分割时 PSRE 算法的 1 矩阵空间和生成时间列出在表 5 中, 其中 $p=1$ 时的两列数据即为RMC 算法的时空复杂度. 从表看出, 当 $p=2, 4$ 时 PSRE 算法明显优于 RMC 算法, 且消除了由于样本数过大引起的"out of memory"现象.

图2 进一步表明是否需要进行数据集分割的准则: 当数据量不是很大时, 不需要分割数据集; 当数据集样本数很大时甚至影响到算法执行时, 必需分割大数据集.

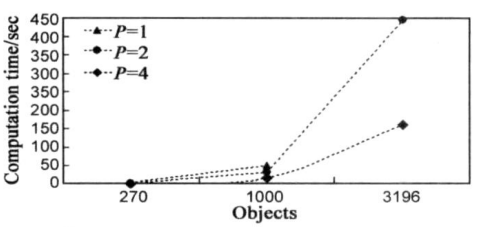

图2　分割数对时间的影响对比

6　结论和下一步的工作

从大数据集甚至海量数据中提取规则是个非常困难的事, 本文在联合决策矩阵算法的基础上, 根据 FMRE 算法存在的问题, 进一步讨论了任意分割策略规则获取算法, 提出了规则获取的串行进位链计算流程, 减少了矩阵的计算和存储, 因此更加适合大数据集上的规则提取, 同时也适应了在增量式的数据中提取规则.

不足之处在于: (1) 由于 PSRE 算法是建立在等价矩阵定义的基础之上, 这种矩阵没有限制同一决策类之间的比较, 因此也无疑增加了矩阵的占用空间和生成时间; (2) 只是考虑了样本数对矩阵算法的影响而采用了(横向) 分割样本的方法而没有考虑(纵向) 分割条件属性数的方法.

针对以上不足, 我们将综合考虑 FMRE 算法和 PSRE 算法各自在规则获取上的优越性, 研究相应的规则提取算法.

参考文献:

[1] Skowron A, Rauszer C. The discernibility matrices and functions in information systems[A]. Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory[C]. Dordrecht: Kluwer Academic Publishers, 1992. 331 – 362.

[2] Hu, X, Cercone, N. Learning in relational database: a rough set approach[J]. Computational Intelligence, 1995, 2: 323 – 337.

[3] 叶东毅,陈昭炯. 一个新的差别矩阵及其求核方法[J]. 电子学报,2002,30(7):1086- 1088.
Ye Dong-yi, Chen Zhao-jiong. A new discernibility matrix and the computation of a core[J]. Acta Electronica Sinica, 2002, 30(7): 1086- 1088. (in Chinese)

[4] 刘少辉,盛秋戬,等. Rough 集高效算法的研究[J]. 计算机学报,2003,26(5):524- 529.
Liu Shao-hui, Sheng Qiu-jian, et al. Research on efficient algorithms for rough set methods[J]. Chinese Journal of Computers, 2003, 26(5): 524- 529. (in Chinese)

[5] 徐章艳,刘作鹏,等. 一个复杂度为 max(O(|C||U|, O(|C|2|U/C|)的快速属性约简算法[J]. 计算机学报,2006,29(3):391- 399.
Xu Zhang-yan, Liu Zuo-peng, et al. A quick attribute reduction algorithm with complexity of max(O(|C||U|, O(|C|2|U/C|)[J]. Chinese Journal of Computers, 2006, 29(3): 391- 399. (in Chinese)

[6] Bazan, G J, Nguyen, H S, et al. Rough set algorithms in classification problem[A]. Rough Set Methods and Applications[C]. Heidelberg: Physica-Verlag, 2000. 49- 88.

[7] 王加阳,陈松乔,罗安. 粗集动态约简研究[J]. 小型微型计算机系统,2006,27(11):2056- 2060.
Wang Jia-yang, Chen Song-qiao, Luo An. Study for dynamic reduct based on rough set[J]. Journal of Chinese Computer Systems, 2006, 27(11): 2056- 2060. (in Chinese)

[8] Krysckiewicz M, Rybinski H. Finding reducts in composed information systems[A]. Rough Sets and Knowledge Discovery (RSKD'93)[C]. Banff: Spring-Verlay, 1993. 259- 268.

[9] Dayong Deng, Houkuan Huang. A new discernibility matrix and function[A]. Rough Set and Knowledge Technology(RSKT'06)[C]. Heidelberg: Springer-Verlag, 2006. 114- 121.

[10] Guan J W, Bell D A, Guan Z. Matrix computation information systems[J]. Information Sciences, 2001, 131(1- 4): 129- 156.

[11] 文香军,蔡云泽,谭天乐,等. 基于粗糙属性向量树的规则提取快速矩阵算法[J]. 电子学报,2006,34(1):65- 70.
Wen Xiang-jun, Cai Yun-ze, Tan Tianle, et al. Fast matrix computation algorithms based on RAVT for rules extraction[J]. Acta Electronica Sinica, 2006, 34(1): 65- 70. (in Chinese)

[12] 谭天乐,宋执环,李平. 信息系统数据清洗、规则提取的矩阵算法[J]. 信息与控制,2003,32(4):289- 294.
Tan Tian-le, Song Zhi-huan, Li Ping. Matrix computation for data cleaning and rule extraction in information system[J]. Information and Control, 2003, 32(4): 289- 294. (in Chinese)

[13] 黄兵,周献中. 不完备信息系统分配约简与规则提取的矩阵算法[J]. 计算机工程,2005,31(17):20- 22.
Huang Bing, Zhou Xian-zhong. Matrix computation for assignment reduction and rule extraction in incomplete information systems[J]. Computer Engineering, 2005, 31(17): 20- 22. (in Chinese)

[14] 程玉胜,张佑生,胡学钢. 大数据集上基于串行进位链规则提取的矩阵分块算法[J]. 中国科学技术大学学报,2009,39(2):196- 203.
Cheng Yu-sheng, Zhang You-sheng, Hu Xue-gang. Matrix block computation with serial carry chain for rule extraction in massive data sets[J]. Journal of University of Science and Technology of China, 2009, 39(2): 196- 203. (in Chinese)

[15] 程玉胜,张佑生,胡学钢. 两类决策系统中规则获取的联合决策矩阵算法[J]. 系统工程理论与实践,2008,28(6):137- 142.
Cheng Yu-sheng, Zhang You-sheng, Hu Xue-gang. Joint decision matrix computation for rules extraction in two classes of decision systems[J]. Systems Engineering-Theory & Practice, 2008, 28(6): 137- 142. (in Chinese)

[16] 程玉胜,张佑生,胡学钢. 不完备决策系统中规则提取的快速矩阵算法[J]. 系统仿真学报,2008,20(15):4036- 4040.
Cheng Yu-sheng, Zhang You-sheng, Hu Xue-gang. Fast matrix computation algorithm for rules extraction in incomplete decision systems[J]. Journal of System Simulation, 2008, 20(15): 4036- 4040. (in Chinese)

作者简介:

程玉胜 男,1969 年生于安徽桐城. 博士,副教授. 研究方向为粗糙集理论与算法、智能信息系统等.
E-mail: chengyshaq@163.com

张佑生 男,1941 年生于湖南浏阳. 博士生导师,教授. 研究方向为人工智能、人工智能、数据挖掘、和计算机图形学等.